**Практическая работа №4**

**Сафиуллин Ринат 320П**

using System;

namespace Practice4

{

internal class Program

{

static void Main(string[] args)

{

Func<double,double> task1Func = x => 2 \* x - 5 \* Math.Log(x, Math.E) - 3;

Func<double, double> task1FuncDerivative = x => -5 / x;

Func<double, double> task7Func = x => 3 \* Math.Sin(8 \* x) - 0.7 \* x + 0.9;

Console.WriteLine($"Задание 4: {MethodChord(task1Func, 0.5, 1, 0.000001)}");

Console.WriteLine($"Задание 5: {MethodTangent(task1Func, task1FuncDerivative, 0.5, 1, 0.000001)}");

Console.WriteLine($"Задание 6: {CombinedMethod(task1Func, task1FuncDerivative, 0.5, 1, 0.000001)}");

Console.WriteLine($"Задание 7: {MethodChord(task7Func, 1.9, 2, 0.000001)}");

}

static double MethodChord(Func<double, double> func, double a, double b, double e)

{

var h = (b - a) / 100;

var y = func(a);

var d = func(a + 2 \* h) - 2 \* func(a + h) + func(a);

double x0;

double c;

if (y \* d < 0)

{

x0 = a;

c = b;

}

else

{

x0 = b;

c = a;

}

var x1 = (x0 \* func(c) - c \* func(x0)) / (func(c) - func(x0));

while (Math.Abs(x1 - x0) > e)

{

x0 = x1;

x1 = (x0 \* func(c) - c \* func(x0)) / (func(c) - func(x0));

}

return x1;

}

static double MethodTangent(Func<double, double> func, Func<double, double> funcDerivative, double a, double b, double e)

{

var h = (b - a) / 100;

var x = func(a);

var y = func(a + 2 \* h) - 2 \* func(a + h) + func(a);

double x0;

if (x \* y > 0)

{

x0 = a;

}

else

{

x0 = b;

}

var x1 = x0 - func(x0)/funcDerivative(x0);

while (Math.Abs(x1 - x0) > e)

{

x0 = x1;

x1 = x0 - func(x0) / funcDerivative(x0);

}

return x1;

}

static double CombinedMethod(Func<double, double> func, Func<double, double> funcDerivative, double a, double b, double e)

{

var h = (b - a) / 100;

var x = func(a);

var y = func(a + 2 \* h) - 2 \* func(a + h) + func(a);

double x0;

double x1;

double c;

if (x \* y < 0)

{

x0 = a;

x1 = b;

c = b;

}

else

{

x0 = b;

x1 = a;

c = a;

}

var x2 = (x0 \* func(c) - c \* func(x0)) / (func(c) - func(x0));

var x3 = x1 - func(x1) / funcDerivative(x1);

while (Math.Abs(x3 - x2) > 2 \* e)

{

x0 = x2;

x2 = (x0 \* func(c) - c \* func(x0)) / (func(c) - func(x0));

x1 = x3;

x3 = x1 - func(x1) / funcDerivative(x1);

}

return (x2 + x3) / 2;

}

}

}